

## Contents

<b>1 Routine/Function Prologues</b>	<b>2</b>
1.0.1 noah_gfrac.F90 (Source File: noah_gfrac.F90) . . . . .	2

## 1 Routine/Function Prologues

### 1.0.1 noah\_gfrac.F90 (Source File: noah\_gfrac.F90)

This subroutine takes vegetation greenness fraction data and the date to interpolate and determine the actual value of the greenness fraction for that date. This actual value is then returned to the main program. The assumption is that the data point is valid for the 16th of the given month, at 00Z.

#### REVISION HISTORY:

28 Apr 2002: K. Arsenault; Added NOAH LSM to LDAS, initial code

#### INTERFACE:

```
subroutine noah_gfrac
```

#### USES:

```
use noah_varder
use time_manager
use time_module
use lisdrv_module, only : grid,tile,lis
#ifndef ( defined OPEN DAP )
  use opendap_module
#endif
```

#### CONTENTS:

```
noahdrv%noah_gflag = 0
zeroi=0
numi=16
!-----
! Determine Monthly data Times (Assume Monthly value valid at DA=16)
!-----
if(lis%da.lt.16)then
  mo1=lis%t%mo-1
  yr1=lis%t%yr
  if(mo1.eq.0)then
    mo1=12
    yr1=lis%t%yr-1
  endif
  mo2=lis%t%mo
  yr2=lis%t%yr
else
  mo1=lis%t%mo
  yr1=lis%t%yr
  mo2=lis%t%mo+1
  yr2=lis%t%yr
  if(mo2.eq.13)then
    mo2=1
```

```

        yr2=lis%t%yr+1
    endif
endif

call date2time(time1,doy1,gmt1,yr1,mo1,&
    numi,zeroi,zeroi,zeroi)
call date2time(time2,doy2,gmt2,yr2,mo2,&
    numi,zeroi,zeroi,zeroi)
!-----
!  Weights to be used to interpolate greenness fraction values.
!-----
wt1= (time2-lis%t%time)/(time2-time1)
wt2= (lis%t%time-time1)/(time2-time1)
!-----
!  Determine if GFRAC files need to be updated
!-----
if(time2 .gt. noahdrv%noah_gfractime) then
    gfrac_flag = 1
else
    gfrac_flag = 0
endif

if(gfrac_flag .eq. 1) then
    noahdrv%noah_gfractime = time2
    noahdrv%noah_gflag = 1
!-----
! Open greenness fraction dataset of months corresponding to
! time1 and time2 for selected LDAS domain and read data.
!-----
    write(mmm1,3) mo1
    write(mm2,3) mo2
3    format(i2.2)

#if ( defined opendap )
    print*, 'msg: noah_gfrac -- retrieving gfrac file ',&
        trim(noahdrv%noah_mgfile)//'gfrac_//mm1//'.bfsa',&
        ',(,iam,)'
    call system("opendap_scripts/getgfrac.pl "//ciam// " // &
        trim(noahdrv%noah_mgfile)//'gfrac_//mm1//'.bfsa'      &
        //" //cparm_slat// " //cparm_nlat                  &
        //" //cparm_wlon// " //cparm_elon// " //mm1)
    print*, 'msg: noah_gfrac -- retrieving gfrac file ', &
        trim(noahdrv%noah_mgfile)//'gfrac_//mm2//'.bfsa',&
        ',(,iam,)'
    call system("opendap_scripts/getgfrac.pl "//ciam// " // &
        trim(noahdrv%noah_mgfile)//'gfrac_//mm2//'.bfsa'      &
        //" //cparm_slat// " //cparm_nlat                  &

```

```

//" "//cparm_wlon//" "//cparm_elon//" "//mm2)
#endif
print*, 'msg: noah_gfrac -- retrieving gfrac file ',&
trim(noahdrv%noah_mgfile)//'gfrac_''//mm1//'.bfsa',&
' (',iam,')
open (10, &
file=trim(noahdrv%noah_mgfile)//'gfrac_''//mm1//'.bfsa', &
status='old', form='unformatted')
print*, 'msg: noah_gfrac -- retrieving gfrac file ', &
trim(noahdrv%noah_mgfile)//'gfrac_''//mm2//'.bfsa',&
' (',iam,')
open (11, &
file=trim(noahdrv%noah_mgfile)//'gfrac_''//mm2//'.bfsa', &
status='old', form='unformatted')

read(10) value1
read(11) value2
close(10)
close(11)

!-----
! Assign MONTHLY vegetation greenness fractions to each tile.
!-----
do i=1,lis%d%nch
  if((value1(tile(i)%col, tile(i)%row-tnroffset) .ne. -9999.000) &
  .and.(value2(tile(i)%col, tile(i)%row-tnroffset).ne.-9999.000)) &
  then
    noah(i)%vegmp1=value1(tile(i)%col, tile(i)%row-tnroffset)
    noah(i)%vegmp2=value2(tile(i)%col, tile(i)%row-tnroffset)
  endif
end do
endif
!-----
! Interpolate greenness fraction values once daily
!-----

if (noahdrv%noah_gfracdchk .ne. lis%t%da) then
  noahdrv%noah_gflag = 1
  do i=1,lis%d%nch
    noah(i)%vegip = (wt1*noah(i)%vegmp1)+(wt2*noah(i)%vegmp2)
  end do
  noahdrv%noah_gfracdchk = lis%t%da
  print*, 'Done noah_gfrac', ' (',iam,')

  if(lis%o%wparam.eq.1) then
    allocate(gfracout(lis%d%lnc,lis%d%lnr))
    do i=1,lis%d%nch
      if(grid(i)%lat*1000.ge.lis%d%kgds(4).and. &

```

```
grid(i)%lat*1000.le.lis%d%kgds(7).and. &
grid(i)%lon*1000.ge.lis%d%kgds(5).and. &
grid(i)%lon*1000.le.lis%d%kgds(8)) then
rindex = tile(i)%row - (lis%d%kgds(4)-lis%d%kgds(44)) &
/lis%d%kgds(9)
cindex = tile(i)%col - (lis%d%kgds(5)-lis%d%kgds(45)) &
/lis%d%kgds(10)
gfracout(cindex,rindex) = noah(i)%vegip*1.0
endif
enddo
open(32,file="gfracout.bin",form='unformatted')
write(32) gfracout
close(32)
deallocate(gfracout)
endif
end if
return
```